Foundation of Cryptography (0368-4162-01), Lecture 4

Pseudorandom Functions

Iftach Haitner, Tel Aviv University

November 29, 2011

Section 1

Function Families

function families

1. $\mathbb{F} = \{\mathbb{F}_n\}_{n\in\mathbb{N}}$, where $\mathbb{F}_n = \{f\colon \{0,1\}^{m(n)} \mapsto \{0,1\}^{\ell(n)}\}$
2. We write $\mathbb{F} = \{\mathbb{F}_n\colon \{0,1\}^{m(n)} \mapsto \{0,1\}^{\ell(n)}\}$
3. If $m(n) = \ell(n) = n$, we omit it from the notation
4. We identify function with their description
5. The rv F_n is uniformly distributed over $\mathbb{F}_n$

efficient function families

Definition 1 (efficient function family)

An ensemble of function families $\mathcal{F} = \{\mathcal{F}_n\}_{n\in\mathbb{N}}$ is efficient, if the following hold:

Samplable. $\mathcal{F}$ is samplable in polynomial-time: there exists a PPT that given 1^n, outputs (the description of) a uniform element in $\mathcal{F}_n$.

Efficient. There exists a polynomial-time algorithm that given $x \in \{0,1\}^n$ and (a description of) $f \in \mathcal{F}_n$, outputs $f(x)$.

random functions

Definition 2 (random functions)

For $m, \ell \in \mathbb{N}$, we let $\Pi_{m,\ell}$ consist of all functions from $\{0,1\}^m$ to $\{0,1\}^\ell$.

random functions

Definition 2 (random functions)

For $m, \ell \in \mathbb{N}$, we let $\Pi_{m,\ell}$ consist of all functions from $\{0,1\}^m$ to $\{0,1\}^\ell$.

- It takes $(2^\ell)^{2^m}$ bits to describe an element inside $\Pi_{m,\ell}$.

random functions

Definition 2 (random functions)

For $m, \ell \in \mathbb{N}$, we let $\Pi_{m,\ell}$ consist of all functions from $\{0,1\}^m$ to $\{0,1\}^\ell$.

- It takes $(2^\ell)^{2^m}$ bits to describe an element inside $\Pi_{m,\ell}$.
- We sometimes think of $\pi \in \Pi_{m,\ell}$ as a random string of length $2^m \cdot \ell$.

random functions

Definition 2 (random functions)

For $m, \ell \in \mathbb{N}$, we let $\Pi_{m,\ell}$ consist of all functions from $\{0,1\}^m$ to $\{0,1\}^\ell$.

- It takes $(2^\ell)^{2^m}$ bits to describe an element inside $\Pi_{m,\ell}$.
- We sometimes think of $\pi \in \Pi_{m,\ell}$ as a random string of length $2^m \cdot \ell$.
- $\Pi_n = \Pi_{n,n}$

pseudorandom functions

Definition 3 (pseudorandom functions)

A function family ensemble $\mathbb{F} = \{\mathbb{F}_n : \{0,1\}^n \mapsto \{0,1\}^{m(n)}\}$ is pseudorandom, if

$$\left|\Pr[\mathsf{D}^{\mathbb{F}_n}(1^n) = 1] - \Pr[\mathsf{D}^{\Pi_{n,m(n)}}(1^n) = 1\right| = \operatorname{neg}(n),$$

for any oracle-aided PPT D.

pseudorandom functions

Definition 3 (pseudorandom functions)

A function family ensemble $\mathbb{F} = \{\mathbb{F}_n : \{0,1\}^n \mapsto \{0,1\}^{m(n)}\}$ is pseudorandom, if

$$\left|\Pr[\mathsf{D}^{\mathbb{F}_n}(1^n) = 1] - \Pr[\mathsf{D}^{\Pi_{n,m(n)}}(1^n) = 1\right| = \operatorname{neg}(n),$$

for any oracle-aided PPT D.

1. Suffices to consider $m(n) = n$

pseudorandom functions

Definition 3 (pseudorandom functions)

A function family ensemble $\mathbb{F} = \{\mathbb{F}_n : \{0,1\}^n \mapsto \{0,1\}^{m(n)}\}$ is pseudorandom, if

$$\left|\Pr[\mathrm{D}^{\mathbb{F}_n}(1^n) = 1] - \Pr[\mathrm{D}^{\Pi_{n,m(n)}}(1^n) = 1\right| = \operatorname{neg}(n),$$

for any oracle-aided PPT D.

1. Suffices to consider $m(n) = n$
2. Easy to construct (with no assumption) for $\Pi^{m,\ell}$ with $m \in O(\log n)$ and $\ell \in \operatorname{poly}(n)$

pseudorandom functions

Definition 3 (pseudorandom functions)

A function family ensemble $\mathbb{F} = \{\mathbb{F}_n : \{0,1\}^n \mapsto \{0,1\}^{m(n)}\}$ is pseudorandom, if

$$\left| \Pr[\mathsf{D}^{\mathbb{F}_n}(1^n) = 1] - \Pr[\mathsf{D}^{\Pi_{n,m(n)}}(1^n) = 1 \right| = \operatorname{neg}(n),$$

for any oracle-aided PPT D.

1. Suffices to consider $m(n) = n$
2. Easy to construct (with no assumption) for $\Pi^{m,\ell}$ with $m \in O(\log n)$ and $\ell \in \operatorname{poly}(n)$
3. PRF easily imply a PRG

pseudorandom functions

Definition 3 (pseudorandom functions)

A function family ensemble $\mathbb{F} = \{\mathbb{F}_n : \{0,1\}^n \mapsto \{0,1\}^{m(n)}\}$ is pseudorandom, if

$$\left|\Pr[\mathsf{D}^{\mathbb{F}_n}(1^n) = 1] - \Pr[\mathsf{D}^{\Pi_{n,m(n)}}(1^n) = 1\right| = \operatorname{neg}(n),$$

for any oracle-aided PPT D.

1. Suffices to consider $m(n) = n$
2. Easy to construct (with no assumption) for $\Pi^{m,\ell}$ with $m \in O(\log n)$ and $\ell \in \operatorname{poly}(n)$
3. PRF easily imply a PRG
4. Pseudorandom permutations (PRPs)

Section 2

PRF from OWF

the construction

Construction 4

Let $g\colon \{0,1\}^n \mapsto \{0,1\}^{2n}$. Let $g_0(s) = g(s)_{1,\dots,n}$ and $g_1(s) = g(s)_{n+1,\dots,2n}$. For s and $x \in \{0,1\}^*$, let f_s be defined as

$$f_s(x) = g_{x_n}(\dots(g_{x_2}(g_{x_1}(s))))$$

the construction

Construction 4

Let $g\colon \{0,1\}^n \mapsto \{0,1\}^{2n}$. Let $g_0(s) = g(s)_{1,\dots,n}$ and $g_1(s) = g(s)_{n+1,\dots,2n}$. For s and $x \in \{0,1\}^*$, let f_s be defined as

$$f_s(x) = g_{x_n}(\dots(g_{x_2}(g_{x_1}(s))))$$

Let $\mathbb{F}_n = \{f_s\colon s \in \{0,1\}^n\}$ and $\mathbb{F} = \{\mathbb{F}_n\}$.

the construction

Construction 4

Let $g\colon \{0,1\}^n \mapsto \{0,1\}^{2n}$. Let $g_0(s) = g(s)_{1,\ldots,n}$ and $g_1(s) = g(s)_{n+1,\ldots,2n}$. For s and $x \in \{0,1\}^*$, let f_s be defined as

$$f_s(x) = g_{x_n}(\ldots(g_{x_2}(g_{x_1}(s))))$$

Let $\mathbb{F}_n = \{f_s\colon s \in \{0,1\}^n\}$ and $\mathbb{F} = \{\mathbb{F}_n\}$.

g is efficient function implies that $\mathbb{F}$ is an efficient family.

the construction

Construction 4

Let $g\colon \{0,1\}^n \mapsto \{0,1\}^{2n}$. Let $g_0(s) = g(s)_{1,\dots,n}$ and $g_1(s) = g(s)_{n+1,\dots,2n}$. For s and $x \in \{0,1\}^*$, let f_s be defined as

$$f_s(x) = g_{x_n}(\dots(g_{x_2}(g_{x_1}(s))))$$

Let $\mathbb{F}_n = \{f_s\colon s \in \{0,1\}^n\}$ and $\mathbb{F} = \{\mathbb{F}_n\}$.

g is efficient function implies that $\mathbb{F}$ is an efficient family.

Theorem 5 (Goldreich-Goldwasser-Micali)

If g is a PRG then $\mathbb{F}$ is a PRF.

the construction

Construction 4

Let $g \colon \{0,1\}^n \mapsto \{0,1\}^{2n}$. Let $g_0(s) = g(s)_{1,\ldots,n}$ and $g_1(s) = g(s)_{n+1,\ldots,2n}$. For s and $x \in \{0,1\}^*$, let f_s be defined as

$$f_s(x) = g_{x_n}(\ldots(g_{x_2}(g_{x_1}(s))))$$

Let $\mathbb{F}_n = \{f_s \colon s \in \{0,1\}^n\}$ and $\mathbb{F} = \{\mathbb{F}_n\}$.

g is efficient function implies that $\mathbb{F}$ is an efficient family.

Theorem 5 (Goldreich-Goldwasser-Micali)

If g is a PRG then $\mathbb{F}$ is a PRF.

Corollary 6

OWFs imply PRFs.

Proof Idea

- For a fixed $s \in \{0,1\}^n$, consider the execution tree T_s

Proof Idea

- For a fixed $s \in \{0,1\}^n$, consider the execution tree T_s
- Observation: $\mathrm{D} = (g(g_0(U_n)), g(g_1(U_n)))$ is pseudorandom:

Proof Idea

- For a fixed $s \in \{0,1\}^n$, consider the execution tree T_s
- Observation: $\mathrm{D} = (g(g_0(U_n)), g(g_1(U_n)))$ is pseudorandom:
  Proof: $\mathrm{D}' = (g(U_n^{(0)}), g(U_n^1)) \approx_c U_{4n}$ and $\mathrm{D} \approx_c \mathrm{D}'$.

Proof Idea

- For a fixed $s \in \{0,1\}^n$, consider the execution tree T_s
- Observation: $\mathrm{D} = (g(g_0(U_n)), g(g_1(U_n)))$ is pseudorandom:
  Proof: $\mathrm{D}' = (g(U_n^{(0)}), g(U_n^1)) \approx_c U_{4n}$ and $\mathrm{D} \approx_c \mathrm{D}'$.
- Hence we can handle input of length 1
- Extend to longer inputs?

Proof Idea

- For a fixed $s \in \{0,1\}^n$, consider the execution tree T_s
- Observation: $\mathrm{D} = (g(g_0(U_n)), g(g_1(U_n)))$ is pseudorandom:
  Proof: $\mathrm{D}' = (g(U_n^{(0)}), g(U_n^{1})) \approx_c U_{4n}$ and $\mathrm{D} \approx_c \mathrm{D}'$.
- Hence we can handle input of length 1
- Extend to longer inputs?
- We show that an efficient sample from the *truth table* of $f \leftarrow \mathbb{F}_n$, is computationally indistinguishable from that of $\pi \leftarrow \Pi_{n,n}$.

Actual proof

Assume $\exists$ PPT D, $p \in$ poly and infinite set $\mathcal{I} \subseteq \mathbb{N}$ with

$$\left|\Pr[\mathsf{D}^{F_n}(1^n) = 1] - \Pr[\mathsf{D}^{\Pi_n}(1^n) = 1]\right| \geq \frac{1}{p(n)}, \qquad (1)$$

for any $n \in \mathcal{I}$ and fix $n \in \mathbb{N}$

Actual proof

Assume $\exists$ PPT D, $p \in$ poly and infinite set $\mathcal{I} \subseteq \mathbb{N}$ with

$$\left|\Pr[\mathrm{D}^{F_n}(1^n) = 1] - \Pr[\mathrm{D}^{\Pi_n}(1^n) = 1]\right| \geq \frac{1}{p(n)}, \qquad (1)$$

for any $n \in \mathcal{I}$ and fix $n \in \mathbb{N}$
Let $t = t(n) \in$ poly be a bound on the running time of $\mathrm{D}(1^n)$.
We use D to construct a PPT D′ such that

$$\left|\Pr[\mathrm{D}'(U_{2n}^t) = 1] - \Pr[\mathrm{D}'(g(U_n)^t) = 1\right| > \delta(n)/n,$$

where $U_{2n}^t = U_{2n}^{(1)}, \ldots, U_{2n}^{(t(n))}$ and
$g(U_n)^t = g(U_n^{(1)}), \ldots, g(U_n^{(t(n))})$.

The hybrid

Let g and f be as in the definition of $\mathbb{F}_n$

Definition 7

For $k \in \{0, \ldots, n\}$, let $\mathcal{H}_k = \{h_\pi \colon \{0,1\}^n \mapsto \{0,1\}^n \colon \pi \in \Pi_{k,n}\}$, where $h_\pi(x) = f_{\pi(x_{1,\ldots,k})}(x_{k+1,\ldots,n})$

The hybrid

Let g and f be as in the definition of $\mathbb{F}_n$

Definition 7

For $k \in \{0, \ldots, n\}$, let $\mathcal{H}_k = \{h_\pi \colon \{0,1\}^n \mapsto \{0,1\}^n \colon \pi \in \Pi_{k,n}\}$, where $h_\pi(x) = f_{\pi(x_{1,\ldots,k})}(x_{k+1,\ldots,n})$

- $f_y(\lambda) = y$
- $\Pi_{0,n} = \{0,1\}^n$, and for $\pi \in \Pi_{0,n}$ let $\pi(\lambda) = \pi$

The hybrid

Let g and f be as in the definition of $\mathbb{F}_n$

Definition 7

For $k \in \{0, \ldots, n\}$, let $\mathcal{H}_k = \{h_\pi \colon \{0,1\}^n \mapsto \{0,1\}^n \colon \pi \in \Pi_{k,n}\}$, where $h_\pi(x) = f_{\pi(x_{1,\ldots,k})}(x_{k+1,\ldots,n})$

- $f_y(\lambda) = y$
- $\Pi_{0,n} = \{0,1\}^n$, and for $\pi \in \Pi_{0,n}$ let $\pi(\lambda) = \pi$

- Note that $\mathcal{H}_0 = \mathbb{F}_n$ and $\mathcal{H}_n = \Pi_{n,n}$

The hybrid

Let g and f be as in the definition of $\mathbb{F}_n$

Definition 7

For $k \in \{0,\ldots,n\}$, let $\mathcal{H}_k = \{h_\pi \colon \{0,1\}^n \mapsto \{0,1\}^n \colon \pi \in \Pi_{k,n}\}$, where $h_\pi(x) = f_{\pi(x_{1,\ldots,k})}(x_{k+1,\ldots,n})$

- $f_y(\lambda) = y$
- $\Pi_{0,n} = \{0,1\}^n$, and for $\pi \in \Pi_{0,n}$ let $\pi(\lambda) = \pi$

- Note that $\mathcal{H}_0 = \mathbb{F}_n$ and $\mathcal{H}_n = \Pi_{n,n}$
- Can we emulate $\mathcal{H}_k$?

The hybrid

Let g and f be as in the definition of $\mathbb{F}_n$

Definition 7

For $k \in \{0,\ldots,n\}$, let $\mathcal{H}_k = \{h_\pi \colon \{0,1\}^n \mapsto \{0,1\}^n \colon \pi \in \Pi_{k,n}\}$, where $h_\pi(x) = f_{\pi(x_{1,\ldots,k})}(x_{k+1,\ldots,n})$

- $f_y(\lambda) = y$
- $\Pi_{0,n} = \{0,1\}^n$, and for $\pi \in \Pi_{0,n}$ let $\pi(\lambda) = \pi$

- Note that $\mathcal{H}_0 = \mathbb{F}_n$ and $\mathcal{H}_n = \Pi_{n,n}$
- Can we emulate $\mathcal{H}_k$? We emulate if from D's point of view.
- We present efficient "function family" $\{\mathcal{O}_k\}_{k\in[n]}$, such that
  - $\mathsf{D}^{O_k(U_{2n}^t)}(1^n) \equiv \mathsf{D}^{H_k}(1^n)$
  - $\mathsf{D}^{O_k(g(U_n)^t)}(1^n) \equiv \mathsf{D}^{H_{k-1}}(1^n)$

  for any $k \in [n]$, where O_k and H_K are uniformly sampled from $\mathcal{O}_k$ and $\mathcal{H}_k$ respectively.

completing the proof

Let $\mathsf{D}'(y)$ return $\mathsf{D}^{O_k(y)}$ for k uniformly chosen in $[n]$.

completing the proof

Let $D'(y)$ return $D^{O_k(y)}$ for k uniformly chosen in $[n]$. Hence

$$\left|\Pr[D'(U_{2n}^t = 1]\right| - \Pr[D'(g(U_n)^t) = 1]$$
$$= \quad \left|\sum_{k=1}^{n} \frac{1}{n} \cdot \Pr[D^{O_k(U_{2n}^t)}(1^n) = 1] - \sum_{k=1}^{n} \frac{1}{n} \cdot \Pr[D^{O_k(g(U_n)^t)}(1^n) = 1]\right|$$

completing the proof

Let $\mathrm{D}'(y)$ return $\mathrm{D}^{O_k(y)}$ for k uniformly chosen in $[n]$. Hence

$$\left|\Pr[\mathrm{D}'(U_{2n}^t = 1]\right| - \Pr[\mathrm{D}'(g(U_n)^t) = 1]$$

$$= \left|\sum_{k=1}^{n} \frac{1}{n} \cdot \Pr[\mathrm{D}^{O_k(U_{2n}^t)}(1^n) = 1] - \sum_{k=1}^{n} \frac{1}{n} \cdot \Pr[\mathrm{D}^{O_k(g(U_n)^t)}(1^n) = 1]\right|$$

$$= \frac{1}{n}\left|\sum_{k=1}^{n} \Pr[\mathrm{D}^{H_k}(1^n) = 1] - \sum_{k=1}^{n} \Pr[\mathrm{D}^{H_{k-1}}(1^n) = 1]\right|$$

completing the proof

Let $\mathrm{D}'(y)$ return $\mathrm{D}^{O_k(y)}$ for k uniformly chosen in $[n]$. Hence

$$\left|\Pr[\mathrm{D}'(U_{2n}^t = 1]\right| - \Pr[\mathrm{D}'(g(U_n)^t) = 1]$$

$$= \left|\sum_{k=1}^{n} \frac{1}{n} \cdot \Pr[\mathrm{D}^{O_k(U_{2n}^t)}(1^n) = 1] - \sum_{k=1}^{n} \frac{1}{n} \cdot \Pr[\mathrm{D}^{O_k(g(U_n)^t)}(1^n) = 1]\right|$$

$$= \frac{1}{n}\left|\sum_{k=1}^{n} \Pr[\mathrm{D}^{H_k}(1^n) = 1] - \sum_{k=1}^{n} \Pr[\mathrm{D}^{H_{k-1}}(1^n) = 1]\right|$$

$$= \frac{1}{n}\left|\Pr[\mathrm{D}^{H_n}(1^n) = 1] - \Pr[\mathrm{D}^{H_0}(1^n) = 1]\right| = \frac{1}{np(n)} \square$$

The family $\mathcal{O}_k$

$$\mathcal{O}_k := \{O_{s^1,\ldots,s^t} : s^1, \ldots, s^t \in \{0,1\}^n \times \{0,1\}^n\}.$$

Algorithm 8 ($O_{s^1,\ldots,s^t}$)

On the i'th query $x^i \in \{0,1\}^n$:

1. If x^ℓ with $x^\ell_{1,\ldots,k-1} = x^i_{1,\ldots,k-1}$ was previously asked, set $z = s^\ell_{x_k}$ (where ℓ is the minimal such index).
   Otherwise, set $z = s^i_{x_k}$.
2. Return $f_z(x_{k+1,\ldots,n})$

The family $\mathcal{O}_k$

$$\mathcal{O}_k := \{O_{s^1,\dots,s^t} : s^1,\dots,s^t \in \{0,1\}^n \times \{0,1\}^n\}.$$

Algorithm 8 ($O_{s^1,\dots,s^t}$)

On the i'th query $x^i \in \{0,1\}^n$:

1. If x^ℓ with $x^\ell_{1,\dots,k-1} = x^i_{1,\dots,k-1}$ was previously asked, set $z = s^\ell_{x_k}$ (where ℓ is the minimal such index). Otherwise, set $z = s^i_{x_k}$.
2. Return $f_z(x_{k+1,\dots,n})$

The "oracle" is *stateful*.

The family $\mathcal{O}_k$

$$\mathcal{O}_k := \{O_{s^1,\ldots,s^t} : s^1,\ldots,s^t \in \{0,1\}^n \times \{0,1\}^n\}.$$

Algorithm 8 ($O_{s^1,\ldots,s^t}$)

On the i'th query $x^i \in \{0,1\}^n$:

1. If x^ℓ with $x^\ell_{1,\ldots,k-1} = x^i_{1,\ldots,k-1}$ was previously asked, set $z = s^\ell_{x_k}$ (where ℓ is the minimal such index). Otherwise, set $z = s^i_{x_k}$.
2. Return $f_z(x_{k+1,\ldots,n})$

The “oracle" is *stateful*.
We need to prove that $\mathrm{D}^{O_k(U^t_{2n})}(1^n) \equiv \mathrm{D}^{H_k}(1^n)$ and $\mathrm{D}^{O_k(g(U_n)^t)}(1^n) \equiv \mathrm{D}^{H_{k-1}}(1^n)$.

$\mathsf{D}^{O_k(U_{2n}^t)}(1^n) \equiv \mathsf{D}^{H_k}(1^n)$

Proposition 9

For any $\ell, m \in \mathbb{N}$ and any algorithm A, it holds that $A^{\Pi_{\ell,m}} \equiv A^{B_{\ell,m}}$, where the stateful random algorithm $B_{\ell,m}$ answers identical queries with the same answer, and answers new queries with a random string of length m.

$\mathrm{D}^{O_k(U^t_{2n})}(1^n) \equiv \mathrm{D}^{H_k}(1^n)$

Proposition 9

For any $\ell, m \in \mathbb{N}$ and any algorithm A, it holds that $A^{\Pi_{\ell,m}} \equiv A^{B_{\ell,m}}$, where the stateful random algorithm $B_{\ell,m}$ answers identical queries with the same answer, and answers new queries with a random string of length m.

Proof?

$D^{O_k(U_{2n}^t)}(1^n) \equiv D^{H_k}(1^n)$

Proposition 9

For any $\ell, m \in \mathbb{N}$ and any algorithm A, it holds that $A^{\Pi_{\ell,m}} \equiv A^{B_{\ell,m}}$, where the stateful random algorithm $B_{\ell,m}$ answers identical queries with the same answer, and answers new queries with a random string of length m.

Proof? Does the above trivialize the whole issue of PRF?

$\mathrm{D}^{O_k(U_{2n}^t)}(1^n) \equiv \mathrm{D}^{H_k}(1^n)$

Proposition 9

For any $\ell, m \in \mathbb{N}$ and any algorithm A, it holds that $A^{\Pi_{\ell,m}} \equiv A^{B_{\ell,m}}$, where the stateful random algorithm $B_{\ell,m}$ answers identical queries with the same answer, and answers new queries with a random string of length m.

Proof? Does the above trivialize the whole issue of PRF?Let $\widetilde{O}_k$ be the variant that returns z (and not $f_{x_{k+1,\ldots,n}}(z)$) and let $\widetilde{\mathrm{D}}_k$ be the algorithm that implements D using $\widetilde{O}_k$ (by computing $f_{x_{k+1,\ldots,n}}(z)$ by itself).

$D^{O_k(U^t_{2n})}(1^n) \equiv D^{H_k}(1^n)$

Proposition 9

For any $\ell, m \in \mathbb{N}$ and any algorithm A, it holds that $A^{\Pi_{\ell,m}} \equiv A^{B_{\ell,m}}$, where the stateful random algorithm $B_{\ell,m}$ answers identical queries with the same answer, and answers new queries with a random string of length m.

Proof? Does the above trivialize the whole issue of PRF?Let $\widetilde{O}_k$ be the variant that returns z (and not $f_{x_{k+1,\ldots,n}}(z)$) and let $\widetilde{\mathrm{D}}_k$ be the algorithm that implements D using $\widetilde{O}_k$ (by computing $f_{x_{k+1,\ldots,n}}(z)$ by itself).
By Proposition 9

$$\mathrm{D}^{O_k(U^t_{2n})}(1^n) \equiv \widetilde{\mathrm{D}}_k^{\widetilde{O}_k(U^t_{2n})}(1^n) \equiv \widetilde{\mathrm{D}}_k^{\pi_{k,n}}(1^n) \equiv \mathrm{D}^{H_k}(1^n) \tag{2}$$

$D^{O_k(g(U_n)^t)}(1^n) \equiv D^{H_{k-1}}(1^n)$

It holds that

$$D^{O_k(g(U_n)^t)}(1^n) \equiv D^{O_{k-1}(U_{2n}^t)}(1^n) \tag{3}$$

$\mathsf{D}^{O_k(g(U_n)^t)}(1^n) \equiv \mathsf{D}^{H_{k-1}}(1^n)$

It holds that

$$\mathsf{D}^{O_k(g(U_n)^t)}(1^n) \equiv \mathsf{D}^{O_{k-1}(U_{2n}^t)}(1^n) \tag{3}$$

Hence, by Equation (2)

$$\mathsf{D}^{O_k(g(U_n)^t)}(1^n) \equiv \mathsf{D}^{H_{k-1}}(1^n)$$

Section 3

PRP from PRF

Pseudorandom permutations

Let $\widetilde{\Pi}_n$ be the set of all permutations over $\{0,1\}^n$.

Definition 10 (pseudorandom permutations)

A permutation ensemble $\mathbb{F} = \{\mathbb{F}_n : \{0,1\}^n \mapsto \{0,1\}^n\}$ is a pseudorandom permutation, if

$$\left|\Pr[\mathsf{D}^{\mathbb{F}_n}(1^n) = 1] - \Pr[\mathsf{D}^{\widetilde{\Pi}_n}(1^n) = 1\right| = \operatorname{neg}(n), \tag{4}$$

for any oracle-aided PPT D

Pseudorandom permutations

Let $\widetilde{\Pi}_n$ be the set of all permutations over $\{0,1\}^n$.

Definition 10 (pseudorandom permutations)

A permutation ensemble $\mathbb{F} = \{\mathbb{F}_n \colon \{0,1\}^n \mapsto \{0,1\}^n\}$ is a pseudorandom permutation, if

$$\left|\Pr[\mathsf{D}^{\mathbb{F}_n}(1^n) = 1] - \Pr[\mathsf{D}^{\widetilde{\Pi}_n}(1^n) = 1\right| = \operatorname{neg}(n), \tag{4}$$

for any oracle-aided PPT D

- Equation (4) holds for any PRF

Construction

Construction 11

Given a function family $\mathbb{F} = \{\mathbb{F}_n \colon \{0,1\}^n \mapsto \{0,1\}^n\}$, let $\mathrm{LR}(\mathbb{F}) = \{\mathrm{LR}(\mathbb{F}_n) \colon \{0,1\}2n \mapsto \{0,1\}^{2n}\}$, where $\mathrm{LR}(\mathbb{F}_n) = \{\mathrm{LR}(f) \colon f \in \mathbb{F}_n\}$ and $\mathrm{LR}(f)(\ell, r) = (r, f(r) \oplus \ell)$.

Construction

Construction 11

Given a function family $\mathbb{F} = \{\mathbb{F}_n \colon \{0,1\}^n \mapsto \{0,1\}^n\}$, let $\mathrm{LR}(\mathbb{F}) = \{\mathrm{LR}(\mathbb{F}_n) \colon \{0,1\}2n \mapsto \{0,1\}^{2n}\}$, where $\mathrm{LR}(\mathbb{F}_n) = \{\mathrm{LR}(f) \colon f \in \mathbb{F}_n\}$ and $\mathrm{LR}(f)(\ell, r) = (r, f(r) \oplus \ell)$.
For $i \in \mathbb{N}$, let $\mathrm{LR}^i(\mathbb{F})$ be the i'th iteration of $\mathrm{LR}(\mathbb{F})$.

Construction

Construction 11

Given a function family $\mathbb{F} = \{\mathbb{F}_n \colon \{0,1\}^n \mapsto \{0,1\}^n\}$, let $\mathrm{LR}(\mathbb{F}) = \{\mathrm{LR}(\mathbb{F}_n) \colon \{0,1\}2n \mapsto \{0,1\}^{2n}\}$, where $\mathrm{LR}(\mathbb{F}_n) = \{\mathrm{LR}(f) \colon f \in \mathbb{F}_n\}$ and $\mathrm{LR}(f)(\ell, r) = (r, f(r) \oplus \ell)$.
For $i \in \mathbb{N}$, let $\mathrm{LR}^i(\mathbb{F})$ be the i'th iteration of $\mathrm{LR}(\mathbb{F})$.

$\mathrm{LR}(\mathbb{F})$ is always a permutation family, and is efficient if $\mathbb{F}$ is.

Construction

Construction 11

Given a function family $\mathbb{F} = \{\mathbb{F}_n \colon \{0,1\}^n \mapsto \{0,1\}^n\}$, let $\mathrm{LR}(\mathbb{F}) = \{\mathrm{LR}(\mathbb{F}_n) \colon \{0,1\}2n \mapsto \{0,1\}^{2n}\}$, where $\mathrm{LR}(\mathbb{F}_n) = \{\mathrm{LR}(f) \colon f \in \mathbb{F}_n\}$ and $\mathrm{LR}(f)(\ell, r) = (r, f(r) \oplus \ell)$.
For $i \in \mathbb{N}$, let $\mathrm{LR}^i(\mathbb{F})$ be the i'th iteration of $\mathrm{LR}(\mathbb{F})$.

$\mathrm{LR}(\mathbb{F})$ is always a permutation family, and is efficient if $\mathbb{F}$ is.

Theorem 12 (Luby-Rackoff)

Assuming that $\mathbb{F}$ is a PRF, then $\mathrm{LR}^3(\mathbb{F})$ is a PRP

Construction

Construction 11

Given a function family $\mathbb{F} = \{\mathbb{F}_n \colon \{0,1\}^n \mapsto \{0,1\}^n\}$, let $\mathrm{LR}(\mathbb{F}) = \{\mathrm{LR}(\mathbb{F}_n) \colon \{0,1\}2n \mapsto \{0,1\}^{2n}\}$, where $\mathrm{LR}(\mathbb{F}_n) = \{\mathrm{LR}(f) \colon f \in \mathbb{F}_n\}$ and $\mathrm{LR}(f)(\ell, r) = (r, f(r) \oplus \ell)$.
For $i \in \mathbb{N}$, let $\mathrm{LR}^i(\mathbb{F})$ be the i'th iteration of $\mathrm{LR}(\mathbb{F})$.

$\mathrm{LR}(\mathbb{F})$ is always a permutation family, and is efficient if $\mathbb{F}$ is.

Theorem 12 (Luby-Rackoff)

Assuming that $\mathbb{F}$ is a PRF, then $\mathrm{LR}^3(\mathbb{F})$ is a PRP

It suffices to prove the the following holds for any $n \in \mathbb{N}$ (why?)

Claim 13

$|\Pr[\mathsf{D}^{\mathrm{LR}^3(\Pi_n)}(1^n) = 1] - \Pr[\mathsf{D}^{\widetilde{\Pi}_{2n}}(1^n)| = 1] \leq \frac{4 \cdot q^2}{2^n}$,
for *any* q-query algorithm D.

Section 4

Applications

general paradigm

Design a scheme assuming that you have random functions, and the realize them using PRF.

Private-key Encryption

Construction 14 (PRF-based encryption)

Given an (efficient) PRF $\mathbb{F}$, define the encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}))$ se:

Key generation $\mathsf{Gen}(1^n)$ returns $k \leftarrow \mathbb{F}_n$

Encryption $\mathsf{Enc}_k(m)$ returns $U_n, k(U_n) \oplus m$

Decryption $\mathsf{Dec}_k(c = (c_1, c_n))$ returns $k(c_1) \oplus c_2$

Private-key Encryption

Construction 14 (PRF-based encryption)

Given an (efficient) PRF $\mathbb{F}$, define the encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$) se:

Key generation $\mathsf{Gen}(1^n)$ returns $k \leftarrow \mathbb{F}_n$

Encryption $\mathsf{Enc}_k(m)$ returns $U_n, k(U_n) \oplus m$

Decryption $\mathsf{Dec}_k(c = (c_1, c_n))$ returns $k(c_1) \oplus c_2$

- Advantages over the PRG based scheme?

Private-key Encryption

Construction 14 (PRF-based encryption)

Given an (efficient) PRF $\mathbb{F}$, define the encryption scheme $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$) se:

Key generation $\mathrm{Gen}(1^n)$ returns $k \leftarrow \mathbb{F}_n$

Encryption $\mathrm{Enc}_k(m)$ returns $U_n, k(U_n) \oplus m$

Decryption $\mathrm{Dec}_k(c = (c_1, c_n))$ returns $k(c_1) \oplus c_2$

- Advantages over the PRG based scheme?
- Proof of security

Message Authentication Code (MAC)

Goal: message authentication.

Message Authentication Code (MAC)

Goal: message authentication.

Definition 15 (MAC)

A MAC is a tuple of PPT's (Gen, Mac, Vrfy) such that

1. $\mathrm{Gen}(1^n)$ outputs a key $k \in \{0,1\}^*$
2. $\mathrm{Mac}(k, m)$ outputs a “tag" t
3. $\mathrm{Vrfy}(k, m, t)$ output 1 (YES) or 0 (NO)

Message Authentication Code (MAC)

Goal: message authentication.

Definition 15 (MAC)

A MAC is a tuple of PPT's (Gen, Mac, Vrfy) such that

1. $\mathsf{Gen}(1^n)$ outputs a key $k \in \{0,1\}^*$
2. $\mathsf{Mac}(k, m)$ outputs a "tag" t
3. $\mathsf{Vrfy}(k, m, t)$ output 1 (YES) or 0 (NO)

We require

Consistency: $\mathsf{Vrfy}(k, m, t) = 1$ for any $k \in \mathsf{Supp}(\mathsf{Gen}(1^n))$, $m \in \{0,1\}^n$ and $t = \mathsf{Mac}(k, m)$

Unforgability: No PPT wins the MAC game with respect to (Gen, Mac, Vrfy)

Definition 16 (MAC game)

Let $(\mathrm{Gen}, \mathrm{Mac}, \mathrm{Vrfy})$ be a MAC and let $K_n = \mathrm{Gen}(1^n)$. An oracle-aided algorithm A wins the MAC game with respect to $(\mathrm{Gen}, \mathrm{Mac}, \mathrm{Vrfy})$, if the following is not negligible:

$$(m, t) \leftarrow A^{\mathrm{Mac}(K_n,\cdot),\mathrm{Vrfy}(K_n,\cdot,\cdot)}(1^n) \wedge \mathrm{Vrfy}(K_n, m, t) = 1$$
$$\wedge\ \mathrm{Mac}(K_n, \cdot) \text{ was not asked on } m$$

Definition 16 (MAC game)

Let (Gen, Mac, Vrfy) be a MAC and let $K_n = \text{Gen}(1^n)$. An oracle-aided algorithm A wins the MAC game with respect to (Gen, Mac, Vrfy), if the following is not negligible:

$$(m,t) \leftarrow A^{\text{Mac}(K_n,\cdot),\text{Vrfy}(K_n,\cdot,\cdot)}(1^n) \wedge \text{Vrfy}(K_n,m,t) = 1$$
$$\wedge\ \text{Mac}(K_n,\cdot) \text{ was not asked on } m$$

- “Private key" definition

Definition 16 (MAC game)

Let (Gen, Mac, Vrfy) be a MAC and let $K_n = \mathrm{Gen}(1^n)$. An oracle-aided algorithm A wins the MAC game with respect to (Gen, Mac, Vrfy), if the following is not negligible:

$$(m,t) \leftarrow A^{\mathrm{Mac}(K_n,\cdot),\mathrm{Vrfy}(K_n,\cdot,\cdot)}(1^n) \wedge \mathrm{Vrfy}(K_n,m,t) = 1$$
$$\wedge\ \mathrm{Mac}(K_n,\cdot) \text{ was not asked on } m$$

- “Private key" definition
- Variable length messages?

Definition 16 (MAC game)

Let (Gen, Mac, Vrfy) be a MAC and let $K_n = \text{Gen}(1^n)$. An oracle-aided algorithm A wins the MAC game with respect to (Gen, Mac, Vrfy), if the following is not negligible:

$$(m,t) \leftarrow A^{\text{Mac}(K_n,\cdot),\text{Vrfy}(K_n,\cdot,\cdot)}(1^n) \wedge \text{Vrfy}(K_n,m,t) = 1$$
$$\wedge\ \text{Mac}(K_n,\cdot) \text{ was not asked on } m$$

- “Private key" definition
- Variable length messages?
- Definition too strong? Any message? Use of Verifier?

Definition 16 (MAC game)

Let (Gen, Mac, Vrfy) be a MAC and let $K_n = \text{Gen}(1^n)$. An oracle-aided algorithm A wins the MAC game with respect to (Gen, Mac, Vrfy), if the following is not negligible:

$$(m, t) \leftarrow A^{\text{Mac}(K_n,\cdot),\text{Vrfy}(K_n,\cdot,\cdot)}(1^n) \wedge \text{Vrfy}(K_n, m, t) = 1$$
$$\wedge\ \text{Mac}(K_n, \cdot) \text{ was not asked on } m$$

- “Private key" definition
- Variable length messages?
- Definition too strong? Any message? Use of Verifier?
- “Reply attacks"

Definition 16 (MAC game)

Let (Gen, Mac, Vrfy) be a MAC and let $K_n = \text{Gen}(1^n)$. An oracle-aided algorithm A wins the MAC game with respect to (Gen, Mac, Vrfy), if the following is not negligible:

$$(m,t) \leftarrow A^{\text{Mac}(K_n,\cdot),\text{Vrfy}(K_n,\cdot,\cdot)}(1^n) \wedge \text{Vrfy}(K_n,m,t) = 1$$
$$\wedge\ \text{Mac}(K_n,\cdot) \text{ was not asked on } m$$

- “Private key" definition
- Variable length messages?
- Definition too strong? Any message? Use of Verifier?
- “Reply attacks"

Definition 17 (ℓ-time MAC)

Same as in Definition 15, but security is only required against ℓ-query adversaries.

constructions

Construction 18 (One-time MAC)

$\mathsf{Gen}(1^n) = U_n$, $\mathsf{Mac}(k, m) = k \oplus m$ and $\mathsf{Vrfy}(k, m, t) = 1$ iff $t = k \oplus m$

constructions

Construction 18 (One-time MAC)

$\mathsf{Gen}(1^n) = U_n$, $\mathsf{Mac}(k, m) = k \oplus m$ and $\mathsf{Vrfy}(k, m, t) = 1$ iff $t = k \oplus m$

Construction 19 ($\ell \in \mathrm{poly}$-time MAC, Stateful)

Use ℓ random strings of length n

constructions

Construction 18 (One-time MAC)

$\mathrm{Gen}(1^n) = U_n$, $\mathrm{Mac}(k, m) = k \oplus m$ and $\mathrm{Vrfy}(k, m, t) = 1$ iff $t = k \oplus m$

Construction 19 ($\ell \in$ poly-time MAC, Stateful)

Use ℓ random strings of length n

Construction 20 ($\ell \in$ poly-time MAC)

$\mathrm{Gen}(1^n)$ return a random member in $\mathcal{H}_n$, where $\mathcal{H} = \{\mathcal{H}_n \colon \{0,1\}^n \mapsto \{0,1\}^n\}$ is an efficient family of ℓ-wise independent hash functions.[a]

Let $\mathrm{Mac}(k, m) = k(m)$, and $\mathrm{Vrfy}(k, m, t) = 1$ iff $t = k(m)$.

[a] For any distinct $x_1, \ldots, x_\ell \in \{0,1\}^n$ and $y_1, \ldots, y_\ell \in \{0,1\}^n$, $\Pr h \leftarrow \mathcal{H}_n[h(x_1) = y_1 \wedge \cdots \wedge h(x_\ell) = y_\ell] = 2^{-\ell n}$.

PRF-based MAC

Construction 21 (PRF-based MAC)

Same as Construction 20, but uses a family of length preserving function $\mathbb{F}$ instead of $\mathcal{H}$.

Claim 22

Assuming that $\mathbb{F}$ is a PRF, then Construction 21 is a (poly-time) MAC.

Proof:

PRF-based MAC

Construction 21 (PRF-based MAC)

Same as Construction 20, but uses a family of length preserving function $\mathbb{F}$ instead of $\mathcal{H}$.

Claim 22

Assuming that $\mathbb{F}$ is a PRF, then Construction 21 is a (poly-time) MAC.

Proof: Easy to prove if $\mathbb{F}$ is a family of random functions. Hence, also holds in case $\mathbb{F}$ is a PRF.$\square$